

KERJA PRAKTIK – KI141330

Pembuatan Aplikasi Pembantu Pengawasan Ujian Online ITS Proctor

**DIREKTORAT PENDIDIKAN INSTITUT TEKNOLOGI
SEPULUH NOPEMBER SURABAYA**

Jalan

Periode: 1 Juni 2020 – 15 Juli 2020

Oleh:

Pembimbing Jurusan

Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D

Pembimbing Lapangan

Bagus Jati Santoso, S.Kom., Ph.D

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2020



KERJA PRAKTIK – KI141330

Pembuatan Aplikasi Pembantu Pengawasan Ujian Online ITSProctor

**DIREKTORAT PENDIDIKAN INSTITUT TEKNOLOGI
SEPULUH NOPEMBER**

Periode: 1 Juni 2020 – 15 Juli 2020

Oleh:

Pembimbing Jurusan

Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D

Pembimbing Lapangan

Bagus Jati Santoso, S.Kom., Ph.D

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2020

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

KERJA PRAKTIK

Pembuatan Aplikasi Pembantu Pengawasan Ujian Online ITSProctor

Oleh:

Philip Antoni Siahaan

05111740000111

Furqan Aliyuddien

05111740000124

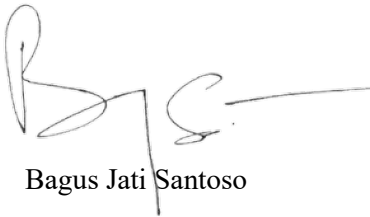
Mengetahui,

Menyetujui,

Dosen Pembimbing

Pembimbing Kerja Praktik

Kerja Praktik



Bagus Jati Santoso

NIP. 198611252018031001



Shintami Chusnul Hidayati

NPP. 1987202012004

SURABAYA

AGUSTUS, 2020

[Halaman ini sengaja dikosongkan]

Pembuatan Aplikasi Pembantu Pengawasan Ujian Online ITSProctor

Nama Mahasiswa : Philip Antoni Siahaan
NRP : 05111740000111
Nama Mahasiswa : Furqan Aliyuddien
NRP : 05111740000124
Departemen : Informatika FTEIC-ITS
**Pembimbing Jurusan : Shintami Chusnul Hidayati, S.Kom.,
M.Sc., Ph.D**
Pembimbing Lapangan : Bagus Jati Santoso, S.Kom., Ph.D

ABSTRAK

Didalam masa pandemi COVID-19, banyak kegiatan-kegiatan akademis yang berubah metode menjadi Online. Dari kegiatan belajar mengajar, seminar online, lomba online, hingga ujian online. Bila meneliti lebih pada kegiatan ujian online, terdapat masalah baru yang timbul yakni apakah ujian tersebut berjalan secara jujur dan nilai dari hasil ujian tersebut dapat dipercaya?

Untuk Menjawab hal tersebut dibutuhkan sebuah aplikasi pengaman pada system ujian sebagai alat bantu dalam mencegah terjadinya kecurangan. Sehingga pengawas ujian dapat terbantu untuk melakukan pengawasan ujian terhadap banyak siswa. Pengawasan dilakukan melalui layar perangkat dengan aplikasi *meeting zoom*. Gambaran umum aplikasi ini adalah dengan melakukan deteksi adanya kecurangan melalui hasil *capture* pada layar dimana, aplikasi telah mengetahui data wajah peserta sebelumnya. Data wajah diolah dengan face recognition untuk mendapatkan apakah terdapat wajah yang sesuai dan deteksi kecurangan. Aplikasi menggunakan Bahasa python dan terhubung dengan database dari pelaksana ujian online. Untuk terhubung dengan data maka dibutuhkan web api untuk dapat bertukar data.

Kata kunci: Python, covid-19, Web Api, Face recognition

KATA PENGANTAR

Puji syukur kami haturkan kepada Allah SWT karena berkat rahmat-Nya kami dapat melaksanakan salah satu kewajiban kami sebagai mahasiswa Departemen Informatika, yakni Kerja Praktek (KP).

Kami menyadari masih ada kekurangan baik dalam pelaksanaan kerja praktik maupun penyusunan buku laporan ini. Namun, kami berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi. Kami mengharapkan kritik dan saran yang membangun untuk kesempurnaan buku laporan kerja praktik ini.

Melalui buku ini, kami juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu, baik langsung maupun tidak langsung, dalam pelaksanaan kerja praktik hingga penyusunan laporan. Orang-orang tersebut antara lain adalah:

1. Kedua orang tua penulis.
2. Ibu Shintami Chusnul Hidayati S.Kom., M.Sc., Ph.D, selaku dosen pembimbing Kerja Praktik.
3. Bapak Ary Mazharuddin Shiddiqi S.Kom., M.Comp., Ph.D, selaku koordinator Kerja Praktik.
4. Bapak Bagus Jati Santoso S.Kom., Ph.D, selaku pembimbing lapangan Kerja Praktik.

Surabaya, Agustus 2020
Philip Antoni Siahaan, Furqan Aliyuddien

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

LEMBAR PENGESAHAN	5
ABSTRAK	8
KATA PENGANTAR	9
DAFTAR ISI	11
BAB I PENDAHULUAN	16
1.1. Latar Belakang	16
1.2. Tujuan	16
1.3. Manfaat	17
1.4. Rumusan Masalah	17
1.5. Lokasi dan Waktu Kerja Praktik	17
1.6. Metodologi Kerja Praktik	17
1.7. Sistematika Laporan	19
<i>[Halaman ini sengaja dikosongkan]</i>	21
BAB II PROFIL PERUSAHAAN	22
2.1. Profil Direktorat Pascasarjana dan Pengembangan Akademik ITS	22
2.2. Logo Perusahaan	24
2.3. Visi Misi Perusahaan	24
2.4. Struktur Organisasi	25
BAB III TINJAUAN PUSTAKA	26
3.1. Sistem Keamanan Ujian Online	26
3.2. API dan Web Service	26
3.3. Antarmuka Pengguna	27
3.4. Aplikasi ITS Proctor	27
BAB IV IMPLEMENTASI SISTEM	31

4.1. Web API	33
4.1.1 Fungsi Index	33
4.1.2 Fungsi Mencari Peserta	34
4.1.3 Fungsi Mencari Peserta Dengan Kode Sesi	35
4.1.4 Fungsi Menyimpan Hasil Capture Aplikasi	36
4.1.5 Fungsi getPhoto	37
4.1.6 Fungsi getByProctor	38
4.1.7 Fungsi StorePic	39
4.2. Proctor App API	40
4.2.1 Fungsi get_all	40
4.2.2 Fungsi get_participant_by_class	41
4.2.3 Fungsi post_report	42
4.2.4 Fungsi getpic_base64	43
4.2.5 Fungsi get_all_pic_by_session	44
4.2.6 Fungsi proctor search	45
4.2.7 Fungsi post_pic	46
4.2.8 Fungsi log_report	47
<i>[Halaman ini sengaja dikosongkan]</i>	48
BAB V PENGUJIAN DAN EVALUASI	49
5.1. Tujuan Pengujian	49
5.2. Kriteria Pengujian	49
5.3. Skenario Pengujian	49
5.4. Hasil Pengujian	49
5.5. Evaluasi Pengujian	56
BAB VI KESIMPULAN DAN SARAN	59

6.1. Kesimpulan	59
6.2. Saran	59
DAFTAR PUSTAKA	61
BIODATA PENULIS I	63
BIODATA PENULIS II	65

DAFTAR GAMBAR

Gambar 2.1 Lambang dan Logo ITS

Gambar 2.2 Struktur Organisasi ITS

Gambar 3.1 UI Aplikasi Proctor

Gambar 3.1 Activity Diagram ITS Proctor

Gambar 4.1 UI Aplikasi Proctor

Gambar 4.2 Code Web API 1

Gambar 4.3 Code Web API 2

Gambar 4.4 Code Web API 3

Gambar 4.5 Code Web API 4

DAFTAR KODE

Kode 4.1 Fungsi Index

Kode 4.2 Route Index

Kode 4.3 Fungsi Mencari Peserta

Kode 4.4 Route get Participant

Kode 4.5 Fungsi Mencari Peserta Dengan Kode Sesi

Kode 4.6 Route get_by_session

Kode 4.7 Fungsi Menyimpan hasil capture aplikasi

Kode 4.8 route Bio_log

Kode 4.9 Fungsi Menyimpan hasil capture aplikasi

Kode 4.10 route get_photo

Kode 4.11 Fungsi getByProctor

Kode 4.12 route get_by_proctor

Kode 4.13 Fungsi StorePic

Kode 4.14 route bio_pic

Kode 4.15 Fungsi get all

Kode 4.16 Fungsi get+participamt by class

Kode 4.17 Fungsi post_report

Kode 4.18 Fungsi post_report

Kode 4.19 Fungsi get_all_pic_by_session

Kode 4.20 Fungsi proctor_search

Kode 4.21 Fungsi post_pic

Kode 4.22 Fungsi log_report

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pandemi yang sedang melanda dunia, termasuk Indonesia saat buku ini ditulis, yakni COVID-19, menyebabkan diberlakukannya pembatasan sosial berskala besar dalam rangka percepatan penanganan penyebaran virus. Pembatasan sosial ini berpengaruh pada pengadaan ujian Seleksi Kemitraan ITS dan Seleksi Mandiri ITS untuk menyeleksi mahasiswa baru angkatan 2020, serta ujian EFL sebagai syarat yudisium mahasiswa yang telah lulus sidang Tugas Akhir. Untuk memfasilitasi pengadaan ketiga ujian tersebut, ujian akan bersifat daring (online), sehingga peserta dapat mengerjakan ujian di rumah masing-masing menggunakan perangkat laptop sehingga ujian dapat berlangsung dengan menaati peraturan pembatasan sosial yang berlaku.

Karena ujian bersifat daring ini, pengamanan peserta ujian butuh perhatian khusus. Sistem ini dibangun untuk membantu pengawas dalam melakukan pengawasan ujian. Sistem ini akan mencocokkan peserta yang hadir saat ujian pada aplikasi Zoom dengan foto pendaftar yang terdaftar di database. Hal ini akan mencegah kecurangan berupa joki ujian. Sistem ini juga mengawasi kehadiran peserta selama ujian dan juga kehadiran pihak lain yang dapat dianggap sebagai kecurangan.

1.2. Tujuan

Tujuan kerja praktik ini adalah untuk menyelesaikan kewajiban kuliah kerja praktik di Institut Teknologi Sepuluh Nopember dengan beban dua SKS. Selain itu juga untuk membangun sistem pengamanan yang dapat digunakan pada ujian yang dilakukan secara Online.

1.3. Manfaat

Dengan adanya sistem yang akan dibuat, diharapkan dapat membantu Institut Teknologi Sepuluh Nopember atau ITS dalam menjaga kejujuran peserta dalam melaksanakan ujian daring.

1.4. Rumusan Masalah

Berikut ini rumusan masalah pada kerja praktik pembuatan ITSProctor:

1. Bagaimana cara membantu pengawas dalam mengawasi ujian online?
2. Bagaimana cara mendeteksi kecurangan didalam ujian online?

1.5. Lokasi dan Waktu Kerja Praktik

Kerja praktik ini dilaksanakan pada waktu dan tempat sebagai berikut:

Lokasi	: Online
Waktu	: 1 Juni 2020 – 30 Juni 2020
Hari Kerja	: Senin - Jumat
Jam Kerja	: 08.00 WIB – 17.00 WIB

1.6. Metodologi Kerja Praktik

Tahapan pengerjaan kerja praktik dapat dijabarkan sebagai berikut:

1. Perumusan Masalah

Aplikasi ujian yang digunakan online yang akan digunakan adalah aplikasi myITS Online Test yang dikembangkan dalam bentuk aplikasi web. Pengawas ujian akan bertatap muka dengan peserta melalui aplikasi zoom meeting yang terintegrasi ke dalam Safe Exam Browser yang akan digunakan oleh peserta. Karena feed

camera peserta akan digunakan oleh zoom, maka pengambilan gambar secara langsung dari camera client tidak dapat dilakukan.

Oleh karena itu, feed gambar didapatkan dari screen capture layar pengawas yang menampilkan zoom meeting.

2. Studi Literatur

Setelah ditentukan rumusan masalah mengenai sistem yang akan dibuat, dilakukan studi literatur mengenai implementasinya. Pada tahap ini dilakukan proses pencarian, pembelajaran, dan pengumpulan informasi yang berkaitan dengan implementasi sistem yang akan dibuat. Informasi dapat diperoleh dari internet ataupun dari proyek sebelumnya yang serupa dan memungkinkan untuk diimplementasikan.

3. Analisis dan Perancangan

Tahap ini meliputi penjelasan mengenai hasil dari studi literatur yang dilakukan. Dari beberapa metode yang ditemukan saat literasi dianalisa metode mana yang paling tepat dan efektif untuk digunakan untuk menyelesaikan permasalahan. Ditentukan bahasa pemrograman yang akan digunakan, batasan-batasan sistem yang dibuat, juga rancangan antarmuka aplikasi yang akan mudah digunakan pengguna.

4. Implementasi Sistem

Pada tahap ini dijelaskan implementasi program yang digunakan pada proses pembuatan sistem keamanan Ujian Online yang akan dibuat. Bagian ini meliputi penjelasan dari pembuatan sistem pengenalan wajah, dan juga proses integrasi aplikasi dengan sistem ujian online, menggunakan API web service, karena sistem ujian online yang dikembangkan berbasis web, dengan framework laravel. Selain itu juga dijelaskan pembuatan antarmuka sistem pengamanan untuk pengguna.

5. Pengujian dan Evaluasi

Pengujian sistem yang dilakukan meliputi pengujian sistem pengenalan wajah, dan juga pengujian komunikasi integrasi dengan sistem ujian online. Pengujian sistem pengenalan wajah dilakukan dengan mencoba beberapa kemungkinan kondisi yang dapat terjadi.

6. Kesimpulan dan Saran

Pada bab ini, dipaparkan kesimpulan yang dapat diambil dan juga saran dalam pengerjaan kerja praktik

1.7. Sistematika Laporan

Laporan kerja praktik ini terdiri dari tujuh bab dengan rincian sebagai berikut:

1. Bab I Pendahuluan

Pada bab ini dijelaskan tentang latar belakang permasalahan, tujuan, waktu pelaksanaan, serta sistematika pengerjaan kerja praktik dan juga penulisan laporan kerja praktik.

2. Bab II Profil Perusahaan

Pada bab ini, dijelaskan secara rinci tentang profil Direktorat Pascasarjana dan Pengembangan Akademik ITS, tempat kami melaksanakan kerja praktik.

3. Bab III Tinjauan Pustaka

Pada bab ini, dijelaskan mengenai tinjauan pustaka dan literatur yang digunakan dalam penyelesaian kerja praktik di Direktorat Pascasarjana dan Pengembangan Akademik ITS.

4. Bab IV Implementasi Sistem

Pada bab ini, berisi penjelasan tahap-tahap yang dilakukan untuk proses pembuatan Aplikasi ITSProctor.

5. Bab V Pengujian dan Evaluasi

Pada bab ini, dijelaskan tentang hasil pengujian dan evaluasi dari sistem yang telah dikembangkan selama pelaksanaan kerja praktik.

6. Bab VI Kesimpulan dan Saran

Pada bab ini, dipaparkan kesimpulan yang dapat diambil dan juga saran selama pengerjaan kerja praktik.

[Halaman ini sengaja dikosongkan]

BAB II

PROFIL PERUSAHAAN

2.1. Profil Direktorat Pascasarjana dan Pengembangan Akademik ITS

Direktorat Pascasarjana dan Pengembangan Akademik merupakan salah satu direktorat di Bawah Wakil rektor Bidang Akademik dan Kemahasiswaan yang dibentuk sesuai Peraturan Rektor ITS No 24 tahun 2019 tentang Organisasi dan Tata Kerja ITS. Di dalam Peraturan Rektor Tersebut, disebutkan pada Pasal 12 bahwa

- (1) Direktorat Pascasarjana dan Pengembangan Akademik mempunyai tugas membantu Wakil Rektor I dalam
 - a. Menyiapkan perumusan dan melaksanakan kebijakan dalam sub bidang kepascasarjanaan dan pengembangan akademik;
 - b. Menyelenggarakan program kerja yang selaras dengan kebijakan dalam sub bidang kepascasarjanaan dan pengembangan akademik;
 - c. Mengevaluasi dan melaporkan kinerja hasil program kerja dalam sub bidang kepascasarjanaan dan pengembangan akademik; dan
 - d. Menyelenggarakan layanan prima dalam sub bidang kepascasarjanaan dan pengembangan akademik sesuai dengan prinsip reformasi birokrasi dan zona integritas.
- (2) Dalam melaksanakan tugas sebagaimana dimaksud pada ayat (1), Direktorat Pascasarjana dan Pengembangan Akademik menyelenggarakan fungsi:

- a. Perumusan perencanaan, tata kelola dan layanan pendidikan untuk program pascasarjana dan profesi;
- b. Penyelenggaraan sistem penerimaan mahasiswa untuk program pascasarjana dan profesi;
- c. Penyelenggaraan kuliah bersama untuk pascasarjana dan profesi;
- d. Penyelenggaraan layanan pendidikan dan beasiswa untuk pascasarjana dan profesi;
- e. Penyelenggaraan pemantauan dan evaluasi pelaksanaan pendidikan untuk pascasarjana dan profesi;
- f. Penyelenggaraan yudisium program pascasarjana dan profesi;
- g. Penyelenggaraan pengelolaan data dan informasi pendidikan untuk program pascasarjana dan profesi;
- h. Pengoordinasian penyiapan akreditasi untuk program pascasarjana dan profesi;
- i. Pengoordinasian pengelolaan program pendidikan internasional untuk program pascasarjana dan profesi;
- j. Perencanaan dan pengembangan kerjasama akademik untuk seluruh jenjang dan jenis program pendidikan;
- k. Perencanaan dan pengelolaan sistem alih kredit dengan perguruan tinggi mitra;
- l. Pengoordinasian mengenai usulan pembukaan, perubahan dan penutupan program studi untuk seluruh jenjang dan jenis program pendidikan;
- m. Perencanaan, pengembangan dan evaluasi kurikulum, baku mutu pendidikan, metode dan teknologi pembelajaran; dan
- n. pelaksanaan evaluasi dan pelaporan penyelenggaraan program kerja sub bidang kepascasarjana dan pengembangan akademik.

2.2. Logo Perusahaan



Gambar 2.1 lambang dan logo its

2.3. Visi Misi Perusahaan

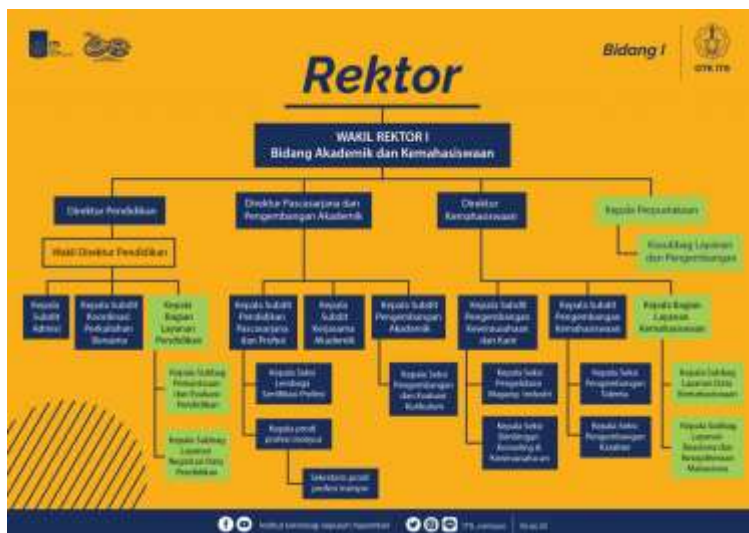
2.3.1. Visi

Menjadi Perguruan Tinggi berkelas dunia yang berkontribusi pada kemandirian bangsa serta menjadi rujukan dalam pendidikan, penelitiandan pengabdian masyarakat serta pengembangan inovasi terutama yang menunjang industri dan kelautan.

2.3.1. Misi

Memberikan kontribusi dalam ilmu pengetahuan dan teknologi untuk kesejahteraan masyarakat melalui kegiatan pendidikan, penelitian, pengabdian kepada masyarakat dan manajemen yang berbasis teknologi informasi dan komunikasi.

2.4. Struktur Organisasi



Gambar 2.2 Struktur Organisasi Direktorat Pascasarjana dan Pengembangan Akademik

BAB III

TINJAUAN PUSTAKA

Pada bab ini, akan dijelaskan mengenai dasar teori yang digunakan selama proses kerja praktik.

3.1. Sistem Keamanan Ujian Online

3.1.1. Pengenalan Wajah

Salah satu hal penting dalam pengawasan ujian adalah memastikan peserta yang mengikuti ujian merupakan peserta yang terdaftar. Untuk itu diperlukan program yang dapat melakukan pengenalan wajah dari data citra. Python merupakan salah satu bahasa pemrograman yang memiliki banyak library opensource. Salah satu library python yang mudah untuk digunakan bernama *face-recognition*. Library tersebut dapat mengenali wajah yang telah dipelajari terlebih dahulu dengan proses yang disebut face encoding.

3.1.2. Deteksi Aplikasi Background

Deteksi Aplikasi background digunakan untuk melihat apakah peserta ujian melakukan kecurangan dengan melakukan pencarian dengan browser maupun menggunakan aplikasi yang dapat membantu dalam pengerjaan soal soal yang diberikan secara online. Aplikasi akan melakukan pendeteksian list aplikasi apa saja yang terbuka saat ujian berlangsung.

3.2. API dan Web Service

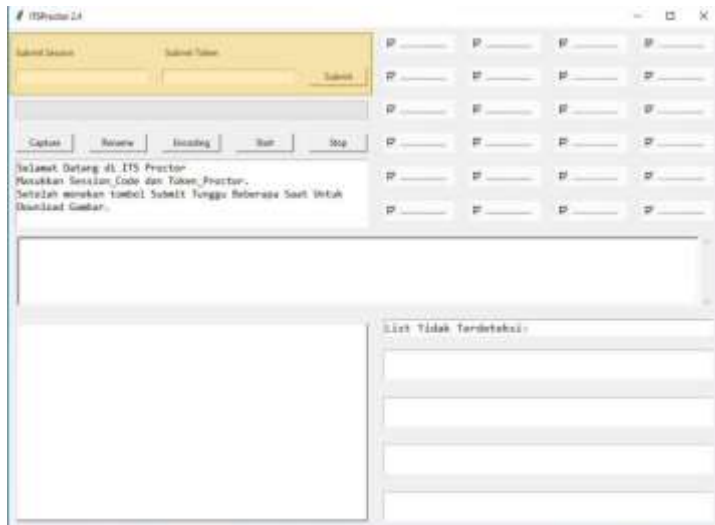
Karena sistem ujian online yang telah dikembangkan berbentuk web apps, dengan framework laravel, diperlukan pembuatan web API pada sistem laravel untuk mendapatkan data peserta ujian dari database sistem ujian online, data yang diambil berupa gambar dan informasi

tentang kelas dan peserta, API ini juga digunakan untuk melakukan menyimpan log hasil pengawasan ke database.

3.3. Antarmuka Pengguna

Karena bahasa pemrograman yang digunakan merupakan python, dibutuhkan library GUI berbasis python yang mudah untuk digunakan. Library GUI open source pada python yang biasa digunakan yaitu meliputi PyQt, Tkinter, PyGUI, dan WxPython. Pada sistem ini digunakan Tkinter karena merupakan salah satu yang mudah penggunaannya. Selain itu digunakan penanda bersuara ketika terjadi keanehan pada saat pemeriksaan wajah peserta.

3.4. Aplikasi ITS Proctor

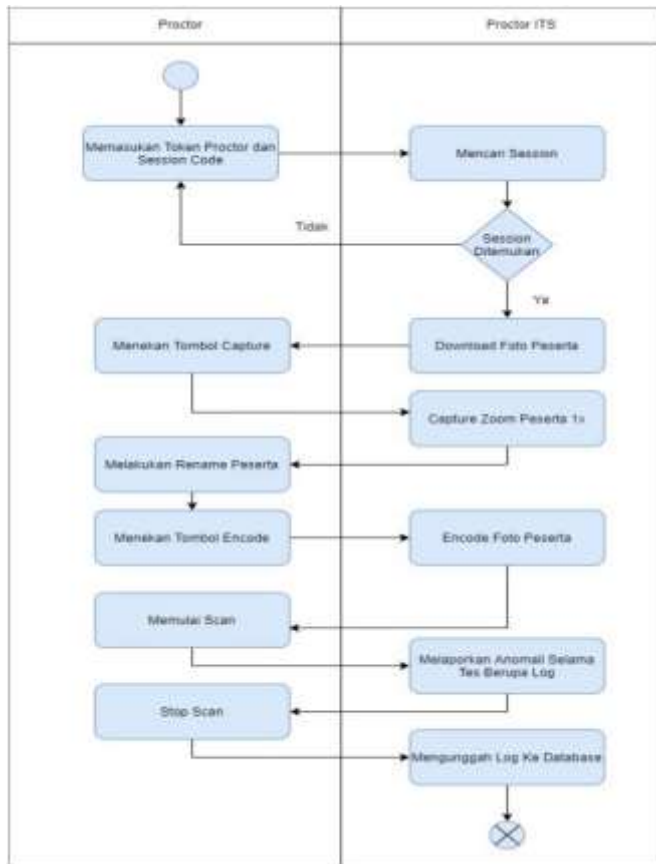


Gambar 3.1 UI Aplikasi Proctor

1.4.1. Deskripsi Singkat

ITS Proctor adalah sebuah perangkat lunak yang dibuat sebagai pembantu untuk pengawas dalam melakukan pengawasan ketika ujian berbasis daring. Perangkat Lunak ini dibuat untuk diberikan kepada DPTSI ITS sebagai bagian dari sistem pengawasan untuk ujian secara daring

1.4.2. Perancangan perangkat lunak



Gambar 3.2 Activity Diagram ITS Proctor

1.4.3. Asset

ITS Proctor adalah sebuah perangkat lunak yang dibuat sebagai pembantu untuk pengawas dalam melakukan pengawasan ketika ujian berbasis daring. Perangkat Lunak ini dibuat untuk diberikan kepada DPTSI ITS sebagai bagian dari sistem pengawasan untuk ujian secara daring.

1.4.4. Sistem

1.4.4.1. Face Recognition

Sistem mendeteksi wajah dari gambar dan mengenali siapa wajah orang tersebut. Dalam pengerjaannya, menggunakan library open source dimana dapat melakukan deteksi wajah dengan hanya menggunakan 1 (satu) foto sebagai training data untuk melakukan deteksi.

1.4.4.2. Screen Capture

Sistem akan mengambil wajah yang nantinya akan dideteksi siapa wajah orang tersebut. Screen Capture dilakukan pada Primary Screen dimana Zoom dinyalakan dan akan mengambil wajah pada Zoom beberapa detik sekali.

Zoom dibuat untuk memuat maksimal 25 wajah tiap screen, dimana wajah akan dideteksi dan diambil ROI nya setiap kali Capture. ROI digunakan untuk dapat melakukan cropping secara otomatis untuk setiap wajah yang tertangkap dalam Primary Screen.

1.4.4.3. Face Matchmaking

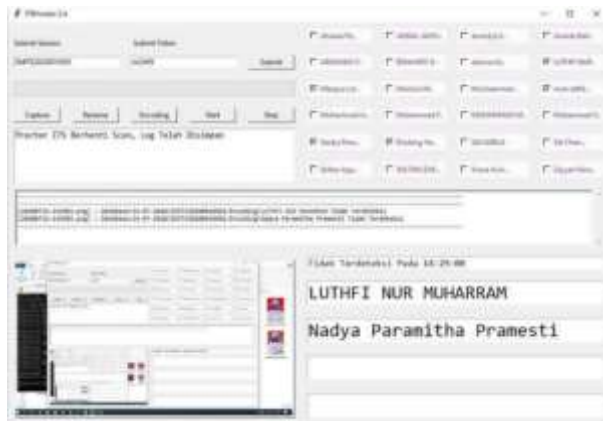
Sistem akan mendeteksi wajah dari gambar dan mengenali siapa wajah. orang tersebut.

1.4.4.4. Report And Logging

Sistem akan melaporkan jika terdapat anomali dan juga memberikan list peserta terduga anomali. Anomali yang dimaksud adalah wajah peserta tidak terdeteksi pada screen capture. Data tersebut nantinya akan di Log dan disimpan menjadi file beserta di upload.

BAB IV IMPLEMENTASI SISTEM

Pada bab ini menjelaskan tahap implementasi yang dilakukan, terdapat 7 fungsi yang dikerjakan dengan semua hasil sudah dapat berjalan sesuai dengan kebutuhan.



Gambar 4.1 UI Aplikasi Proctor



Gambar 4.2 Code Web API 1

```

40 public boolean shouldRespond(Request request) {
41     boolean = (request.getHeader("Authorization") != null);
42
43     if (request.isGet()) {
44         return response().contentType("text/plain").send(200);
45     }
46     else if (request.is("POST")) {
47         return response().contentType("text/plain").send("Spring token authentication", 200);
48     }
49 }
50
51 @Override public boolean isAuth() {
52     return true;
53 }
54
55 public boolean authenticate(Request request, Response response) {
56     boolean = (request.getHeader("Authorization") != null);
57
58     if (request.isGet()) {
59         return response().contentType("text/plain").send(200);
60     }
61     else if (request.is("POST")) {
62         return response().contentType("text/plain").send("Spring token authentication", 200);
63     }
64 }
65
66 @Override public boolean isAuth() {
67     return true;
68 }
69
70 public boolean authenticate(Request request, Response response) {
71     boolean = (request.getHeader("Authorization") != null);
72     return (request.getHeader("Authorization") != null);
73 }
74 }

```

Gambar 4.3 Code Web API 2

```

40 public boolean shouldRespond(Request request) {
41     boolean = (request.getHeader("Authorization") != null);
42
43     if (request.isGet()) {
44         return response().contentType("text/plain").send(200);
45     }
46     else if (request.is("POST")) {
47         return response().contentType("text/plain").send("Spring token authentication", 200);
48     }
49 }
50
51 @Override public boolean isAuth() {
52     return true;
53 }
54
55 public boolean authenticate(Request request, Response response) {
56     boolean = (request.getHeader("Authorization") != null);
57
58     if (request.isGet()) {
59         return response().contentType("text/plain").send(200);
60     }
61     else if (request.is("POST")) {
62         return response().contentType("text/plain").send("Spring token authentication", 200);
63     }
64 }
65
66 @Override public boolean isAuth() {
67     return true;
68 }
69
70 public boolean authenticate(Request request, Response response) {
71     boolean = (request.getHeader("Authorization") != null);
72     return (request.getHeader("Authorization") != null);
73 }
74 }

```

Gambar 4.4 Code Web API 3


```

100 public Function getParticipant(peserta_token, Request: Requester)
101     token = Requester.Headers("token")
102
103     If(token = null)
104         return Response().ToJson(["message" => "No token"], 400)
105     Else If(token != "abc123")
106         return Response().ToJson(["message" => "Wrong token authentication"], 400)
107     Else
108         db := GetDB("project_database")
109         result := db.Query("select * from participant where token = ?",
110             peserta_token, db.PrepareToken)
111     End If
112
113     public Function LoadAll(Request: Requester)
114         token = Requester.Headers("token")
115
116         If(token = null)
117             return Response().ToJson(["message" => "No token"], 400)
118         Else If(token != "abc123")
119             return Response().ToJson(["message" => "Wrong token authentication"], 400)
120         Else
121             db := GetDB("project_database")
122             result := db.Query("select * from participant")
123         End If
124
125         result := result.ToJson()
126         return Response().ToJson(result, 200)
127     End Function
128 End Module

```

Gambar 4.5 Code Web API 4

4.1. Web API

4.1.1 Fungsi Index

4.1.1.1 Deskripsi

Membuat fungsi Mengembalikan seluruh nama peserta ujian pada hari tersebut yang diinput di aplikasi yang dikirimkan melalui request get.

4.1.1.2 Parameter

Request yang berisi app token

4.1.1.3 Data

Data yang digunakan untuk melihat seluruh peserta dan dipakai untuk ditampilkan di list peserta di aplikasi.

4.1.1.4 Source Code

Source code untuk fungsi ini dapat dilihat pada kode 4.1

```

public function index(Request $requests)
{
    $token = $requests->header('apptoken');

    if($token == null){
        return response()->json(['message' => "No token"], 401);
    }
    else if($token != "abc123"){
        return response()->json(['message' => "Wrong token authentication"], 401);
    }

    return Participant::all();
}

```

Kode 4.1 Fungsi Index

4.1.1.5 Route

Source code untuk fungsi ini dapat dilihat pada kode 4.2

```

Route::get('get_participant', 'API\BiometricAPIController@index');

```

Kode 4.2 Route Index

4.1.2 Fungsi Mencari Peserta

4.1.2.1 Deskripsi

Membuat fungsi Mengembalikan nama peserta yang ingin dicari berdasarkan id peserta.

4.1.2.2 Parameter

Request yang berisi app token dan id peserta.

4.1.2.3 Data

Data yang digunakan untuk melihat seluruh peserta dan dipakai untuk ditampilkan di list peserta di aplikasi.

4.1.2.4 Source Code

Source code untuk fungsi ini dapat dilihat pada kode 4.3

```

public function show($id, Request $requests)
{
    $token = $requests->header('apptoken');

    if($token == null){
        return response()->json(['message' => "No token"], 401);
    }
    else if($token != "abc123"){
        return response()->json(['message' => "Wrong token authentication"], 401);
    }

    return Participant::find($id);
}

```

Kode 4.3 Fungsi Mencari Peserta

4.1.2.5 Route

Source code untuk fungsi ini dapat dilihat pada kode 4.4

```

Route::get('get_participant/{id}', 'API\BiometricApiController@show');

```

Kode 4.4 Route get Participant

4.1.3 Fungsi Mencari Peserta Dengan Kode Sesi

4.1.3.1 Deskripsi

Membuat fungsi Mengembalikan identitas peserta yang ingin dicari berdasarkan kode sesi kelas.

4.1.3.2 Parameter

Request yang berisi app token dan session code.

4.1.3.3 Data

Data yang digunakan untuk melihat seluruh peserta dan dipakai untuk ditampilkan di list peserta di aplikasi.

4.1.3.4 Source Code

Source code untuk fungsi ini dapat dilihat pada kode 4.5

```

public function getBySession($session_code, Request $requests){
    $token = $requests->header('apptoken');

    if($token == null){
        return response()->json(['message' => "No token"], 401);
    }
    else if($token != "abc123"){
        return response()->json(['message' => "Wrong token authentication"], 401);
    }

    return DB::table('participant')
    ->select('participant.id_participant','participant.no_participant','participant.name','participant.photo')
    ->where('participant.session_code','=', $session_code)->get();
}

```

Kode 4.3 Fungsi Mencari Peserta Dengan Kode Sesi

4.1.3.5 Route

Source code untuk fungsi ini dapat dilihat pada kode 4.6

```

Route::get('get_by_session/{room_id}', 'API\BiometricAPIController@getBySession');

```

Kode 4.6 Route get_by_session

4.1.4 Fungsi Menyimpan Hasil Capture Aplikasi

4.1.4.1 Deskripsi

Membuat fungsi Menyimpan hasil capture yang berisi data data dari participant

4.1.4.2 Parameter

Request yang berisi app token.

4.1.4.3 Data

Data disimpan untuk menjadi bukti ketika ada nya kegagalan pada ujian yang diambil oleh aplikasi proctor.

4.1.4.4 Source Code

Source code untuk fungsi ini dapat dilihat pada kode 4.7

```

public function store(Request $requests){
    $token = $requests->header('apptoken');

    if($token == null){
        return response()->json(['message' => "No token"], 401);
    }
    else if($token != "abc123"){
        return response()->json(['message' => "Wrong token authentication"], 401);
    }

    $Biolog = Biolog::create([
        'no_participant' => $requests->no_participant,
        'photo'          => $requests->photo, // $path,
        'face_num'       => $requests->face_num,
        'flag'           => $requests->flag,
    ]);

    return response()->json($Biolog, 201);
}

```

Kode 4.7 Fungsi Menyimpan hasil capture aplikasi

4.1.4.5 Route

Source code untuk fungsi ini dapat dilihat pada kode 4.8

```

Route::post('bio_log', 'API\BiometricAPIController@store');

```

Kode 4.8 route Bio_log

4.1.5 Fungsi getPhoto

4.1.5.1 Deskripsi

Membuat fungsi untuk mengambil pas foto milik peserta ujian, pas foto diambil berdasarkan id

4.1.5.2 Parameter

Request yang berisi app token dan id peserta

4.1.5.3 Data

Data disimpan untuk menjadi alat pengecekan apakah peserta yang sesuai dengan pas foto gambar hadir di zoom meeting.

4.1.5.4 Source Code

Source code untuk fungsi ini dapat dilihat pada kode 4.9

```

public function getPhoto($id, Request $requests)
{
    $token = $requests->header('apptoken');

    if($token == null){
        return response()->json(['message' => "No token"], 401);
    }
    else if($token != "abc123"){
        return response()->json(['message' => "Wrong token authentication"], 401);
    }

    $path = null;
    $participant = Participant::find($id);

    if($participant->photo){
        $path=$participant->photo;
    }
    else{
        $path.='9j/4AAQSkZJRgABAQEASABIAAD';
    }

    return response()->json(['name' => 'success', 'data' => $path], 200);
}

```

Kode 4.9 Fungsi Menyimpan hasil capture aplikasi

4.1.5.5 Route

Source code untuk fungsi ini dapat dilihat pada kode 4.10

```

Route::get('get_photo/{id}', 'API\BiometricAPIController@getPhoto');

```

Kode 4.10 route get_photo

4.1.6 Fungsi getByProctor

4.1.6.1 Deskripsi

Membuat fungsi untuk mengambil kode proctor dari sesi yang berjalan, dan menyesuaikan dengan id session yang dimasukan di aplikasi.

4.1.6.2 Parameter

Request yang berisi app token dan id session

4.1.6.3 Data

Data yang diinput dari aplikasi harus sama dengan session yang sesuai dengan id proctor

4.1.6.4 Source Code

Source code untuk fungsi ini dapat dilihat pada kode 4.11

```
public function getByProctor($proctor_token, Request $requests){
    $token = $requests->header('apptoken');

    if($token == null){
        return response()->json(['message' => "No token"], 401);
    }
    else if($token != "abc123"){
        return response()->json(['message' => "Wrong token authentication"], 401);
    }

    return DB::table('proctor_session')
        ->select('proctor_session.session_code')
        ->where('proctor_session.proctor_token','=', $proctor_token)->get();
}
```

Kode 4.11 Fungsi getByProctor

4.1.6.5 Route

Source code untuk fungsi ini dapat dilihat pada kode 4.12

```
#route::get('get_by_proctor/{proctor_token}', 'API\BiometricAPIController@getByProctor');
```

Kode 4.12 route get_by_proctor

4.1.7 Fungsi StorePic

4.1.7.1 Deskripsi

Membuat fungsi untuk menyimpan hasil capture log yang berisi id capture dan path dari foto yang disimpan di database

4.1.7.2 Parameter

Request yang berisi app token

4.1.7.3 Data

Data yang diinput dari aplikasi bisa merupakan base64 atau path dari posisi gambar di database

4.1.7.4 Source Code

Source code untuk fungsi ini dapat dilihat pada kode 4.13

```

public function storePic(Request $requests){
    $token = $requests->header('apptoken');

    if($token == null){
        return response()->json(['message' => "No token"], 401);
    }
    else if($token != "abc123"){
        return response()->json(['message' => "Wrong token authentication"], 401);
    }

    $path = Storage::putFile('public/BioLogs', $requests->file('upload_file'));

    $BioPic = BioPic::create([
        'photo_id' => $requests->photo_id,
        'photo' => $path, // $requests->photo, //
    ]);

    return response()->json($BioPic, 201);
}

```

Kode 4.13 Fungsi StorePic

4.1.7.5 Route

Source code untuk fungsi ini dapat dilihat pada kode 4.12

```

Route::post('bio_pic', 'API\BiometricAPIController@storePic');

```

Kode 4.14 route bio_pic

4.2. Procotor App API

4.2.1 Fungsi get_all

4.2.1.1 Deskripsi

Membuat fungsi pada procotor untuk dapat mengambil data seluruh partisipan dengan melakukan request link metode get.

4.2.1.2 Parameter

-

4.2.1.3 Data

Data diterima dalam bentuk json hasil request kemudian di decode dan di return isi dari data dan hasil request code.

4.2.1.4 Source Code

```
def get_all():#sudah dicoba
    global headers
    #print (headers)
    link = _url('/get_participant')
    request = Request(link, method='GET',headers = headers)
    with urllib.request.urlopen(request) as url:
        data = json.loads(url.read().decode())
    return requests.get(_url('/get_participant'),headers = headers),data
```

Kode 4.15 Fungsi get all

4.2.2 Fungsi get_participant_by_class

4.2.2.1 Deskripsi

Membuat fungsi pada procotor untuk dapat mengambil data peserta proctor berdasarkan id session class yang ingin dicari.

4.2.2.2 Parameter

Id session kelas

4.2.2.3 Data

Data diterima dalam bentuk json hasil request kemudian di decode dan di return isi dari data dan hasil request code.

4.2.2.4 Source Code

```
def get_participant_by_class(id):
    global headers
    link = _url('/get_by_session/'+str(id))
    request = Request(link, method='GET', headers = headers)
    with urllib.request.urlopen(request) as url:
        data = json.loads(url.read().decode())
    return requests.get(_url('/get_by_session/{:s}'.format(str(id))), headers=headers), data
```

Kode 4.16 Fungsi get+participamt by class

4.2.3 Fungsi post_report

4.2.3.1 Deskripsi

Membuat fungsi pada procotor untuk dapat melakukan request post untuk menyimpan data log peserta yang berisi, id peserta, time stamp, banyaknya wajah yang terdeteksi pada satu foto log, kemudian path dari foto tersebut

4.2.3.2 Parameter

id_peserta, foto, time_stamp, jumlah_wajah, flag

4.2.3.3 Data

Data dikirim dalam bentuk json hasil request kemudian di decode dan tidak ada return.

4.2.3.4 Source Code

```
def post_report(id_peserta,foto,time_stamp,jumlah_wajah,flag):
    global headers
    # files = {'upload_file':open(foto,'rb')}
    foto_path = foto
    return requests.post(_url('/bio_log/'), data={
        'no_participant': str(id_peserta),
        'timestamp':time_stamp,
        'face_num': int(jumlah_wajah),
        'flag' : int(flag),
        'photo' : str(foto_path)
    },headers=headers)
```

Kode 4.17 Fungsi post_report

4.2.4 Fungsi getpic_base64

4.2.4.1 Deskripsi

Membuat fungsi pada procotor untuk dapat melakukan request get untuk mengambil data foto dari para peserta ujian. Kemudian disimpan didalam local pc proctor ujian sesuai dengan parameter path. Dalam mengambil foto menggunakan id_peserta dikarenakan akses fungsi hanya untuk satu foto.

4.2.4.2 Parameter

id_peserta, nama file, dan path

4.2.4.3 Data

Data berbentuk json kemudian diterima, disini foto berupa string base 64 yang harus di decode kembali dengan fungsi base64.decode dan disimpan dalam bentuk jpg

4.2.4.4 Source Code

```

def getpic_base64(id_peserta, namafile, path):
    global headers
    global base64_strngg

    resp = requests.get(_url('/get_photo/{}'.format(id_peserta)), headers=headers)
    #andaile null dan format data:image/jpeg,
    base64_string = resp.json()["data"]
    base64_image_str = base64_string[base64_string.find(",")+1:]

    if base64_image_str == None :
        return resp

    try:
        imgdata = base64.b64decode(str(base64_image_str))
    except base64.binascii.Error as err:
        print("incorrect Binary")
        return err

    imgdata = base64.b64decode(str(base64_image_str))
    location = str(path) + '/' + str(namafile)
    with open(location, 'wb') as f:
        f.write(imgdata)
    return resp

```

Kode 4.18 Fungsi post_report

4.2.5 Fungsi get_all_pic_by_session

4.2.5.1 Deskripsi

Membuat fungsi pada procotor untuk dapat melakukan request get untuk mengambil seluruh data foto peserta berdasarkan kelas atau sesi yang sedang dipilih. Kemudian disimpan di path yang telah ditentukan dari aplikasi.

4.2.5.2 Parameter

Session_id, base_path

4.2.5.3 Data

Data peserta diambil dari fungsi get_participant_by class kemudian diarahkan ke fungsi getpic untuk mengambil file gambar.

4.2.5.4 Source Code

```
def get_all_pic_by_session(session_id, base_path):
    global headers
    dates = datetime.today().strftime('%d-%m-%Y')
    count = 0
    try:
        path = "database/" + dates + "/" + str(session_id) + ".json"
        if not os.path.exists(path):
            os.makedirs(path)
        # os.mkdir(path)
    except OSError as error:
        print(error)

    resp_data = get_participant_by_class(session_id)

    # print(data)

    if resp.status_code != 200:
        return 'Cannot fetch items: {}'.format(resp.status_code)
    else:
        for test in data:
            print(test['name'])
            print('{} {} {}'.format(test['id_participant'], test['name'], test['photo']))
            getpic_path = os.path.join(test['id_participant'], "{}.jpg".format(test['name']), test['photo'], base_path)
            count = count + 1
            if count == 24:
                break
        return 'succeed'
```

Kode 4.19 Fungsi get_all_pic_by_session

4.2.6 Fungsi proctor search

4.2.6.1 Deskripsi

Membuat fungsi pada proctor untuk dapat melakukan request get untuk melakukan check terhadap token proctor yang diinput di aplikasi sebagai salah satu requirement login aplikasi.

4.2.6.2 Parameter

Proctor_token

4.2.6.3 Data

Data berupa session kelas yang sedang dijalankan oleh token proctor yang bersangkutan, untuk disesuaikan dengan input session kelas dari aplikasi

4.2.6.4 Source Code

```
def proctor_search(proctor_token):
    global headers
    link = _url('/get_by_proctor/'+str(proctor_token))
    request = Request(link, method='GET', headers = headers)
    with urllib.request.urlopen(request) as url:
        data = json.loads(url.read().decode())
    return requests.get(_url('/get_by_proctor/{:s}'.format(proctor_token)), headers=headers, data
```

Kode 4.20 Fungsi proctor_search

4.2.7 Fungsi post_pic

4.2.7.1 Deskripsi

Membuat fungsi pada proctor untuk dapat melakukan request post untuk mengupload foto ke database dengan melampirkan file foto dan id foto

4.2.7.2 Parameter

Photo, photo id

4.2.7.3 Data

Data berupa gambar yang akan langsung dipost ke api web untuk disimpan dengan nama file dan base path yang sudah diatur di web api.

4.2.7.4 Source Code

```
def post_pic(photo, photo_id):
    files = {'upload_file': open(photo, 'rb')}
    return requests.post(_url('/bio_pic/'), data={
        'photo_id': photo_id
    }, files=files, headers=headers)
```

Kode 4.21 Fungsi post_pic

4.2.8 Fungsi log_report

4.2.8.1 Deskripsi

Membuat fungsi pada procotor untuk dapat melakukan lanjutan request post untuk melakukan penyimpanan terhadap data data log dari aplikasi proctor.

4.2.8.2 Parameter

Id_peserta, foto, timestamp, jumlahwajah, flag

4.2.8.3 Data

Data berupa gambar dan data pendukung launnya yang kemudian diarahkan ke fungsi post_report. Dan disinn data yang di cek adalah respon status dan apakah api nya error

4.2.8.4 Source Code

```
def log_report(id_peserta,foto,time_stamp,jumlah_wajah,flag):
    try:
        resp = post_report(id_peserta,foto,time_stamp,jumlah_wajah,flag)
        if resp.status_code != 201:
            raise ApiError('Cannot post log: {}'.format(resp.status_code))
        else:
            print("data tersimpan")
    except ApiError as error:
        print(error)
    pass
```

Kode 4.22 Fungsi log_report

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba dilakukan terhadap fungsi-fungsi API dan juga Aplikasi yang sudah dibuat.

5.1. Tujuan Pengujian

Pengujian dilakukan terhadap fungsi API agar fungsi-fungsi tersebut dapat mengembalikan informasi yang sesuai dengan yang diinginkan dan dibutuhkan oleh aplikasi. Pengujian terhadap aplikasi ditujukan untuk melihat kinerja aplikasi yang telah dibuat, apakah berjalan sesuai dengan yang diharapkan dan dibutuhkan.

5.2. Kriteria Pengujian

Penilaian atas pencapaian tujuan pengujian API didapatkan dengan memperhatikan kesesuaian output dengan tujuan fungsi tersebut. Sedangkan untuk pengujian aplikasi didapatkan dengan memperhatikan kecepatan pemrosesan dan pengalaman pengguna.

5.3. Skenario Pengujian

Terdapat beberapa skenario pengujian, yaitu:

1. Pengujian fungsi web API dilakukan dengan melihat kesesuaian kebutuhan aplikasi dengan hasil output dari pemanggilan fungsi-fungsi yang ada. Pengujian akan dilakukan dengan menggunakan aplikasi Postman.
2. Pengujian API aplikasi dilakukan dengan melihat output yang didapat menggunakan script python.
3. Pengujian aplikasi dilakukan dengan menghitung waktu tiap proses dijalankan dan juga melihat respon tampilan aplikasi.

5.4. Hasil Pengujian

Terdapat hasil dari code yang telah dibuat pada Bab IV sebelumnya:

5.4.1. Fungsi Web API

5.4.1.1. Fungsi Index

[illegible]

Gambar 5.1 output fungsi index

5.4.1.2. Fungsi Mencari Peserta

```
1  GET https://exam.its.ac.id/api/get_participant/5
2  {
3    "id_participant": 5,
4    "session_code": "SMITS2020010101",
5    "prev_session": null,
6    "package_id": 14,
7    "no_participant": "8200100002",
8    "name": "Muhammad Fahmi Kurniawan",
9    "email": "██████████",
10   "email_sent": null,
11   "email_result": null,
12   "token": "██████████",
13   "session_id": null,
14   "score": 0,
15   "score_section_1": null,
16   "score_section_2": null,
17   "score_section_3": null,
18   "status": 0,
19   "start_time": null,
20   "reason": null,
21   "report": null,
22   "reschedule_report": null,
23   "photo": "https://registrasi.its.ac.id/api/foto/██████████",
24   "reset": 0,
25   "created_at": "2020-06-30 15:01:57",
26   "updated_at": "2020-06-30 15:36:38"
27 }
```

Gambar 5.2 output fungsi mencari peserta

5.4.1.3. Fungsi Mencari Peserta dengan Kode Sesi

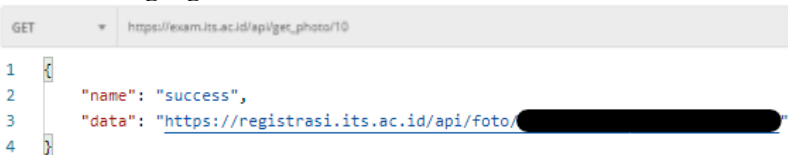


Gambar 5.3 output fungsi mencari peserta dengan id sesi

5.4.1.4. Fungsi Menyimpan Hasil Capture Aplikasi

Karena fungsi ini membutuhkan data untuk di post, maka pengujian tidak bisa dilakukan menggunakan postman. Untuk pengujiannya akan disertakan pada fungsi 5.4.2.3.

5.4.1.5. Fungsi getPhoto



Gambar 5.4 output fungsi get photo

5.4.1.6. Fungsi getByProctor



```

1  [
2      {
3          "session_code": "SMITS20200010202"
4      }
5  ]

```

Gambar 5.5 output fungsi get by proctor

5.4.1.7. Fungsi storePic

Karena fungsi ini membutuhkan data untuk di post, maka pengujian tidak bisa dilakukan menggunakan postman. Untuk pengujiannya akan disertakan pada fungsi 5.4.2.7.

5.4.2. Fungsi APP API

5.4.2.1. Fungsi get all

```

test = ac.get_all()
print(test[0])
print(len(test[1]))
print(test[1][0])

```

Kode 5.1 Pengujian get_all

```

$ python test.py
<Response [200]>
6997
{'id_participant': 1, 'session_code': 'SMITS20200010101', 'prev_session': None, 'package_id': 14, 'n
o_participant': '8260188883', 'name': 'Moch. Syahrul Ramadan', 'email': '
', 'email_sent': None, 'email_result': None, 'token': '
', 'session_id': None, 'score': 0, 'sc
ore_section_1': None, 'score_section_2': None, 'score_section_3': None, 'status': 1, 'start_time': N
one, 'reason': None, 'report': None, 'reschedule_report': None, 'photo': 'https://registrasi.its.ac.
id/api/foto/
', 'reset': 0, 'created_at': '2020-06-30 15:01:57', 'updated_at
': '2020-07-23 01:10:35'}

```

Gambar 5.6 output pengujian get_all

5.4.2.2. Fungsi get participant by class

```

def ambil_kelas(id_class):
    resp, data = ac.get_participant_by_class(id_class)
    if resp.status_code != 200:
        raise ApiError('Cannot fetch all item: {}'.format(resp.status_code))
    else :
        print(data)

```

Kode 5.2 Pengujian get_participant_by_class

```
$ python test.py
[{"id_participant": 11, "no_participant": "8200200025", "name": "Rizky Khairul Waritsin", "photo": "https://registrasi.its.ac.id/api/foto/"}, {"id_participant": 14, "no_participant": "8200200026", "name": "Farah Qanilah Ardiana Arwanto", "photo": "https://registrasi.its.ac.id/api/foto/"}, {"id_participant": 15, "no_participant": "8200200035", "name": "Farisulqisthi Mediawan", "photo": "https://registrasi.its.ac.id/api/foto/"}, {"id_participant": 16, "no_participant": "8200200034", "name": "Daisy Helder Pemas", "photo": "https://registrasi.its.ac.id/api/foto/"}, {"id_participant": 17, "no_participant": "8200200030", "name": "Calvin Matthew Tuwo", "photo": "https://registrasi.its.ac.id/api/foto/"}, {"id_participant": 18, "no_participant": "8200200029", "name": "Rizky Rasadnan S aputra", "photo": "https://registrasi.its.ac.id/api/foto/"}, {"id_participant": 19, "no_participant": "8200200031", "name": "Dimas Wahidono Putra", "photo": "https://registrasi.its.ac.id/api/foto/"}, {"id_participant": 20, "no_participant": "8200200028", "name": "SEPTIANI IDILA SUGIAETO", "photo": "https://registrasi.its.ac.id/api/foto/"}, {"id_participant": 22, "no_participant": "8200200033", "name": "Hendi Rizki Prihantoro", "photo": "https://registrasi.its.ac.id/api/foto/"}
```

Gambar 5.7 Output pengujian get_participant_by_class

5.4.2.3. Fungsi post_report

Pengujian fungsi post_report akan disertakan pada pengujian fungsi 5.4.2.8.

5.4.2.4. Fungsi getpic_base64

```
def save_photo(id_peserta, path):
    try:
        os.mkdir(str(path))
    except OSError as error:
        print(error)
    resp = ac.getpic_base64(id_peserta, "{}.jpg".format(id_peserta), path)
    print(resp)
```

Kode 5.3 Pengujian getpic_base64

```
$ python test.py
[winError: 183] Cannot create a file when that file already exists: 'C:\\\\Users\\\\Furynd\\\\Desktop'
<Response [200]>
```



Gambar 5.8 Output pengujian getpic_base64

5.4.2.5. Fungsi get_all_pic_bysession

```
def all_session_photo(session_id, base_path):
    resp = ac.get_all_pic_bysession(session_id, base_path)
    print(resp)
```

Kode 5.4 Pengujian get_all_pic_bysession

```
(base) C:\Users\Furynd\Desktop\KP\Proctor_ITS_EFL>python test.py
Rizqy Khoirul Waritsin
False
Photo Already In Local Database Folder
Farah Qanita Ardiana Arwanto
False
Farisulqisthi Mediawan
False
Dzaky Haidar Mumtaz
False
Calvin Matthew Tuwo
False
Rizky Ramadhan Saputra
False
Dimas Wibisono Putra
False
SEPTIAINI DELA SUBIAKTO
False
hendi rizki prihantoro
False
saved
```



Gambar 5.9 Output pengujian get_all_pic_bysession

5.4.2.6. Fungsi proctor_search

```
def proctor_hari_sesi_kelas(proctor_token):
    resp, data = ac.proctor_search(proctor_token)
    if resp.status_code != 200:
        raise ApiError('Cannot fetch all item: {}'.format(resp.status_code))
    else:
        print(data[0]['session_code'])
```

Kode 5.5 Pengujian proctor_search

```
$ python test.py  
SMITS20200050116
```

Gambar 5.10 Output pengujian proctor_search

5.4.2.7. Fungsi post_pic

```
def post_photo(path):  
    for root, dirs, files in os.walk(path):  
        path_foto = []  
        for filename in files:  
            path_foto = path + "\\" + str(filename)  
            print(path_foto)  
            resp = ac.post_pic(path_foto,filename)  
            print(resp)
```

Kode 5.6 Pengujian post_pic

```
(base) C:\Users\Furynd\Desktop\KP\Proctor_ITS_EFI>python test.py  
C:\Users\Furynd\Desktop\KP\Database\26-06-2020\sesiTP5D20200104\Foto  
[[]]  
C:\Users\Furynd\Desktop\KP\Database\26-06-2020\sesiTP5D20200104\Foto\Ammar.jpg  
<Response [201]>  
C:\Users\Furynd\Desktop\KP\Database\26-06-2020\sesiTP5D20200104\Foto\Excel 2.jpg  
<Response [201]>  
C:\Users\Furynd\Desktop\KP\Database\26-06-2020\sesiTP5D20200104\Foto\Furqan.jpg  
<Response [201]>  
C:\Users\Furynd\Desktop\KP\Database\26-06-2020\sesiTP5D20200104\Foto\harits.jpg  
<Response [201]>  
C:\Users\Furynd\Desktop\KP\Database\26-06-2020\sesiTP5D20200104\Foto\Hazimi.jpg  
<Response [201]>  
C:\Users\Furynd\Desktop\KP\Database\26-06-2020\sesiTP5D20200104\Foto\Philip.jpg  
<Response [201]>
```

id	photo_id	photo	created_at	updated_at
1	Ammar.jpg	public\BisLogs\MB4FAAt1h43S5y6xxcpqis0IHSS8caabQCK...	2020-10-15 14:17:37	2020-10-15 14:17:37
2	Excel 2.jpg	public\BisLogs\mbXSCrB2Y6thvntUpWjBfMMD04DhGe...	2020-10-15 14:17:38	2020-10-15 14:17:38
3	Furqan.jpg	public\BisLogs\fuHIVXUlmW985J2XSz4E1w2wA7DKw7UQ6tC...	2020-10-15 14:17:39	2020-10-15 14:17:39
4	harits.jpg	public\BisLogs\77JIGnbRcJQTJrHfUj0tghUpCNwclandT...	2020-10-15 14:17:39	2020-10-15 14:17:39
5	Hazimi.jpg	public\BisLogs\ICggtw00w74hBDSbuR8TEukTVz2RHjdgds...	2020-10-15 14:17:40	2020-10-15 14:17:40
6	Philip.jpg	public\BisLogs\M01R5T030Yxp1a795sX75nS4kZbVRnC3x...	2020-10-15 14:17:40	2020-10-15 14:17:40

Gambar 5.11 Output pengujian post_pic

5.4.2.8. Fungsi log_report

```
def log_report(id_session,foto,time_stamp,jumlah_wajah,flag):
    resp = ac.post_report(id_session,foto,time_stamp,jumlah_wajah,flag)
    if resp.status_code != 201:
        raise ApiError('Cannot fetch all item: {}'.format(resp.status_code))
    else:
        print("data tersimpan")
```

Kode 5.7 Pengujian log_report

```
(base) C:\Users\Furynd\Desktop\KP\Proctor_ITS_EFL>python test.py
data tersimpan
```

id	no_participant	photo	face_num	flag	created_at	updated_at
1	1	C:\Users\Furynd\Desktop\KP\Database\26-06-2020\ses...	2	1	2020-10-15 14:21:30	2020-10-15 14:21:30

Gambar 5.12 Output pengujian log_report

5.4.3. Aplikasi Pengenalan Wajah

5.5. Evaluasi Pengujian

Hasil evaluasi pengujian fungsi API dapat dilihat pada tabel 5.1.

Tabel 5.1 Hasil Evaluasi Pengujian Fungsi API

No.	Poin Pengujian	Hasil Pengujian
5.4.1.1	Fungsi Index	Terpenuhi
5.4.1.2	Fungsi Mencari Peserta	Terpenuhi
5.4.1.3	Mencari Peserta dengan Kode Sesi	Terpenuhi
5.4.1.4	Menyimpan Hasil Capture	Terpenuhi
5.4.1.5	Fungsi getPhoto	Terpenuhi
5.4.1.6	Fungsi getByProctor	Terpenuhi
5.4.1.7	Fungsi StorePic	Terpenuhi
5.4.2.1	Fungsi get all	Terpenuhi
5.4.2.2	Fungsi get participant by class	Terpenuhi
5.4.2.3	Fungsi post report	Terpenuhi
5.4.2.4	Fungsi getpic base64	Terpenuhi
5.4.2.5	Fungsi get all pic bysession	Terpenuhi

5.4.2.6	Fungsi proctor search	Terpenuhi
5.4.2.7	Fungsi Post_pic	Terpenuhi

Dengan hasil pengujian pada tabel di atas, dapat disimpulkan bahwa secara keseluruhan fungsi API telah memenuhi kriteria-kriteria yang sudah disebutkan pada bagian-bagian sebelumnya.

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Kesimpulan yang didapat setelah melakukan pembuatan dashboard pada kegiatan kerja praktik di PT. Telekomunikasi Indonesia adalah sebagai berikut:

- Dengan adanya dashboard ini. Para karyawan menjadi mudah untuk melihat kinerja pencapaian kerja mereka selama sebulan dan memudahkan Analyst Manager agar tidak perlu menghitung manual menggunakan kalkulator atau pun excel
- Bahasa pemrograman database yang digunakan oleh PT.Telekomunikas Indonesia saat ini adalah Oracle

6.2. Saran

Saran untuk pengembangan sistem aplikasi Pelaporan Online adalah sebagai berikut:

- Dilakukan pembaruan data pada database setiap ada perubahan karyawan. Agar data bisa selalu ter update

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- Evolve Machine Learner. 2019. Mengenal Machine Learning. <https://medium.com/evolve-machine-learners/mengenal-machine-learning-6c4a48db48b0>.
- Expert System Team. 2020. What is Machine Learning? A Definition. <https://www.expert.ai/blog/machine-learning-definition>.
- Expert System Team. 2020. What is Machine Learning? A Definition. <https://www.expert.ai/blog/machine-learning-definition>.
- Kevin Cristella. 2018. Basic Token Based API Authentication with Laravel. <https://medium.com/@kevincristella/basic-token-based-api-authentication-with-laravel-aeed0050dd0d>.
- Gergő D. Nagy. 2018. Simple JSON Web Token Guard for Laravel APIs. <https://pineco.de/simple-json-web-token-guard-for-laravel-apis>.
- Esther Vaati. 2017. Cara Menggunakan Web API Restful dengan Python. <https://code.tutsplus.com/id/articles/how-to-use-restful-web-apis-in-python--cms-29493>.
- Tedo Haris Chandra. 2018. Build and Testing REST API dengan Aplikasi CRUD Laravel. <https://medium.com/@tedoharischandra29/membangun-dan-testing-rest-api-dengan-crud-sederhana-laravel-687a7d96ab3b>
- Nicesnippets. 2020. Laravel Base64 Image Upload. <https://www.nicesnippets.com/blog/laravel-base64-image-upload>

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS I

Nama : Philip Antoni Siahaan
Tempat, Tanggal Lahir : Pematangsiantar, 17 Maret 1999
Jenis Kelamin : Laki-laki
Agama : Kristen
Status : Belum Menikah
Alamat Asal : Jln Meranti no 70-S, Pematangsiantar,
Sumatera Utara
Alamat Surabaya : Jln Keputih Perintis 4 No 7A, Sukolilo,
Surabaya
Telepon : 085156122109
Email : siahaan.philip279@gmail.com

PENDIDIKAN FORMAL

2017 – sekarang : S1 Informatika ITS
2014 – 2017 : SMA DEL Laguboti
2011 – 2014 : SMP Bintang Timur Pematangsiantar
2005 – 2011 : SD Kalam Kudus Pematangsiantar

KEMAMPUAN

- *Web Programming* (HTML, PHP, CSS, Javascript)
- *Programming* (C, C++, Python)
- *Database Manajemen* (MySQL)
- *Software* Perkantoran (Microsoft Word, Excel, PowerPoint)
- Bahasa (Indonesia, Inggris)

AKADEMIS

Kuliah : Departemen Informatika, Fakultas Teknologi
Informasi dan Komunikasi, Institut Teknologi
Sepuluh Nopember Surabaya
Angkatan : 2017
Semester : 7 (Tujuh)
IPK : 3.52

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS II

Nama : Furqan Aliyuddien
Tempat, Tanggal Lahir : Bedford, 3 Juni 1998
Jenis Kelamin : Laki-laki
Agama : Islam
Status : Belum menikah
Alamat Asal : Jl. H. Junaedi no.13 Cipagalo Girang,
Buah Batu, Kota Bandung
Alamat Surabaya : Jalan Teknik Kimia ITS, Sukolilo, Kota
Surabaya
Telepon : 087824059335
Email : furqanaliyuddien@gmail.com

PENDIDIKAN FORMAL

2015 – sekarang : S1 Informatika ITS
2012 – 2015 : SMA Negeri 8 Bandung
2009 – 2012 : MTs Zakaria
2003 – 2009 : MI Zakaria

KEMAMPUAN

- *Web Programming* (HTML, PHP, CSS, Javascript)
- *Programming* (C, C++, Python)
- *Database Management* (MySQL)
- *Software Perkantoran* (Microsoft Word, Excel, PowerPoint)
- Bahasa (Indonesia, Inggris)

AKADEMIS

Kuliah : Departemen Informatika – Fakultas Teknologi
Informasi dan Komunikasi, Institut Teknologi
Sepuluh Nopember Surabaya
Angkatan : 2017
Semester : 7 (Tujuh)
IPK : 3.70 (6 semester)

[Halaman ini sengaja dikosongkan]